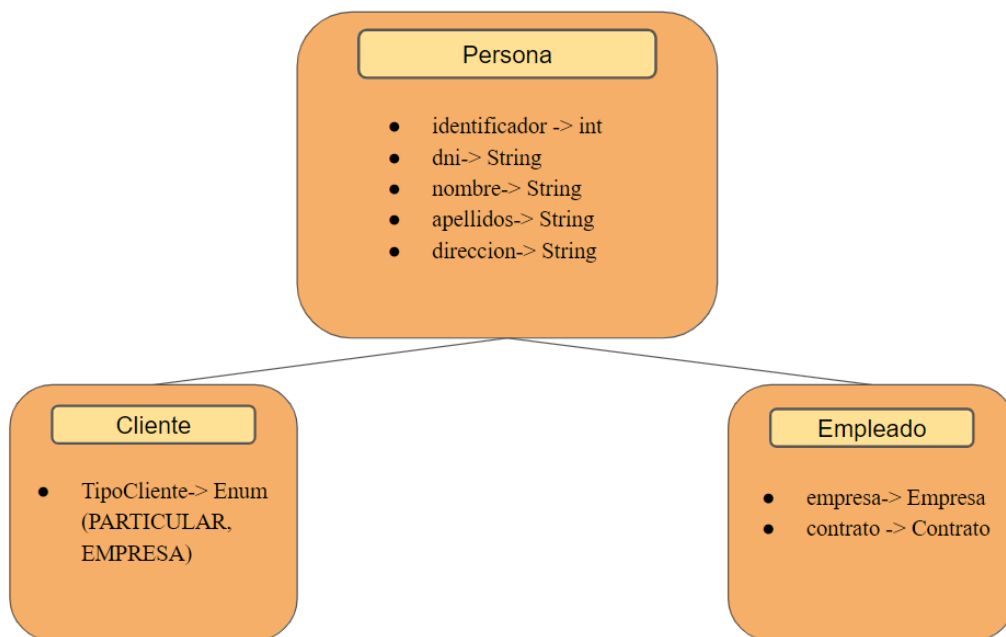


Requisitos V4

1. En la carpeta modelos se ha de implementar la siguiente herencia:



Nota: Esto te puede ayudar.

- Fíjate que los atributos comunes se indican en la superclase , y los atributos específicos se dejan en las subclases.
- No te olvides de poner en las clases hijas el “extends” en la cabecera de la clase.
- Modifica todos los métodos y constructores según hemos visto que tienen que ir indicados en las herencias.
- Si todo ha ido bien, los métodos que tenías implementados que implican a Cliente o Empleado deberían seguir funcionando sin dar fallos.



2. En la clase UtilidadesEmpresa hay que crear los siguientes métodos:

- a. `public List<Empleado> getEmpleadosPymePracticas(List<Empresas> empresas).`

Que devuelve la lista de empleados pertenecientes de las empresas PYMES que se pasan cuyo contrato es del tipo PRACTICAS.

- b. `public Map<Empresa,Empleado> getLosMejorPagadosPorEmpresa(List<Empresa> empresas).`

Que devuelva un mapa con el empleado que más cobra de cada empresa.

- c. `public Map<TipoEmpresa, Integer> getNumEmpleadosPorTipoEmpresa(List<Empresa> empresas).`

Que devuelva un mapa con el número de empleados por cada tipo de empresa.

3. En la clase UtilidadesFactura hay que crear los siguientes métodos:

- a. `public Double gastoTotalCliente(List<Factura> facturas, Cliente cliente).`

Que devuelve la suma del importe total de las facturas , que pertenecen al cliente pasado como parámetro.

- b. `public Double totalFacturadoPeriodo(List<Factura> facturas, LocalDate fecha Inicio, LocalDate fechaFin).`

Que devuelve la suma del importe total de las facturas , que pertenecen al cliente pasado como parámetro.



4. En la clase UtilidadesContrato hay que crear los siguientes métodos:

- a. public Empleado contratarTrabajador(Empresa e, String dni, String nombre, String apellidos, String direccion, String telefono, TipoContrato tipo, Double salario).

Que a partir de los datos que se le pasan genera un empleado asignándole empresa y contrato. EL método tiene que generar los identificadores automáticamente, siendo cada identificador creado diferente.

5. En la clase UtilidadesAlmacen hay que crear los siguientes métodos:

- a. public boolean validarAlmacenes(List<Producto> productos).

El método devuelve true si la suma de los productos que hay en un almacén no supera la capacidad del almacén, teniendo en cuenta todos los productos y todos los almacenes, cada uno con su correspondiente.